Project Report

515111910078 杨俊晨

1. Main idea:

Because the k-th order markov model in which a state contains 1 sequence is equivalent to 1 order markov in which a state contains k sequences, I calculate the transition probability matrix (maximum likelihood) of different k sequences for each genome. Then for each read, I scan the k sequences and calculate the corresponding probability score for each genome matrix and select the maximum score and corresponding genome as the assignment(if the maximum score - the second / second score < 0.05, this read will not be assigned).

2. Test:

I use the given 20000 reads as test to see the performance of my codes. I use different k(ranging from 3 to 11) and compare the assignment result with the seq map(golden standard).

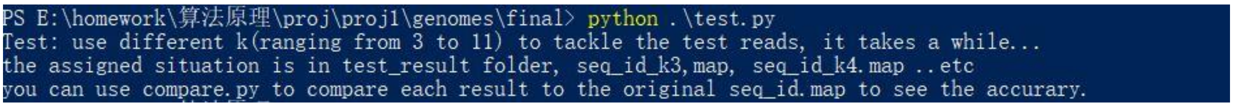
Usage: Command: python test.py

Command: python compare.py

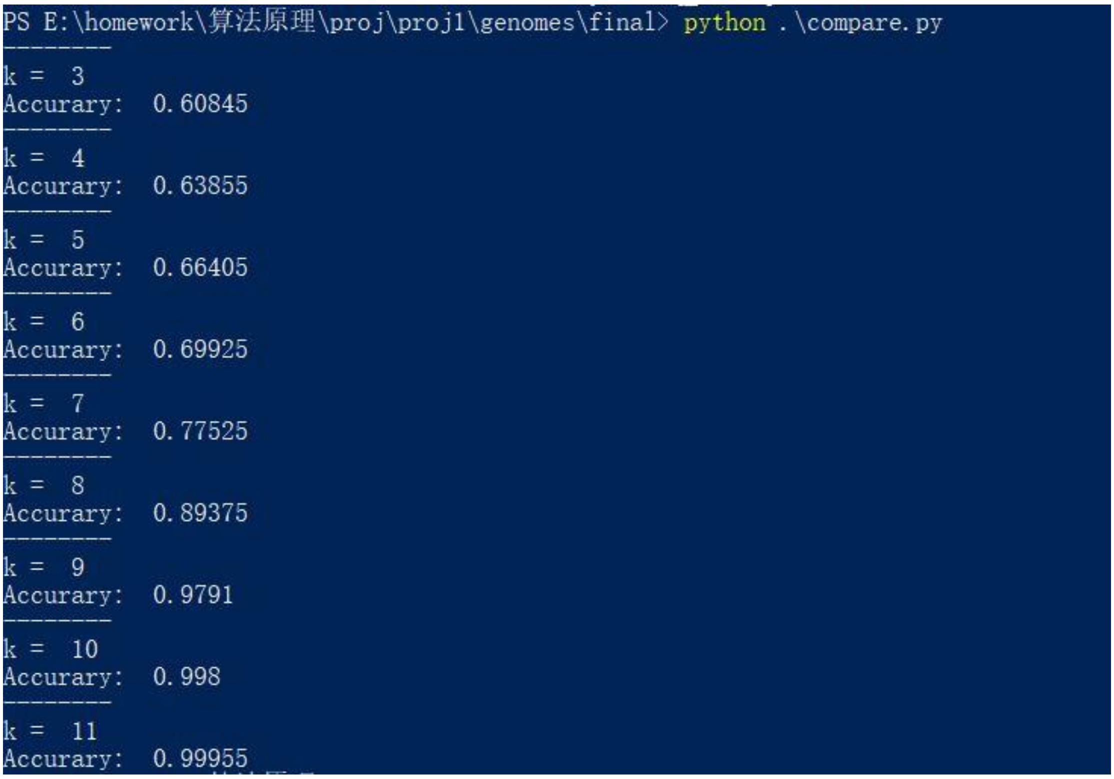
The assigned map for different k will be in test_result folder. Then run compare.py to see the accurary

Note:1. test.py can take 5 minutes to process different k 2. I rename the original genome data file to 0.fna, 1.fna, 2.fna, … ,9.fna)

Here is the comparison results.



```
PS E:\homework\算法原理\proj\proj1\genomes\final> python .\test.py
Test: use different k(ranging from 3 to 11) to tackle the test reads, it takes a while...
the assigned situation is in test_result folder, seq_id_k3,map, seq_id_k4.map ..etc
you can use compare.py to compare each result to the original seq_id.map to see the accurary.
```

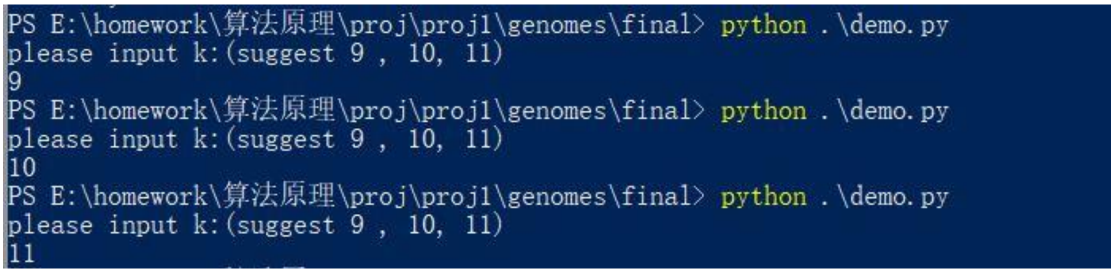


```
PS E:\homework\算法原理\proj\proj1\genomes\final> python .\compare.py
--------
k =  3
Accurary:  0.60845
--------
k =  4
Accurary:  0.63855
--------
k =  5
Accurary:  0.66405
--------
k =  6
Accurary:  0.69925
--------
k =  7
Accurary:  0.77525
--------
k =  8
Accurary:  0.89375
--------
k =  9
Accurary:  0.9791
--------
k =  10
Accurary:  0.998
--------
k =  11
Accurary:  0.99955
```

We can see the bigger the k, the more accurate the results are. So I use k = 9, 10, 11 to demonstrate the final results.

3. Demo:

Command:



```
PS E:\homework\算法原理\proj\proj1\genomes\final> python .\demo.py
please input k:(suggest 9 , 10, 11)
9
PS E:\homework\算法原理\proj\proj1\genomes\final> python .\demo.py
please input k:(suggest 9 , 10, 11)
10
PS E:\homework\算法原理\proj\proj1\genomes\final> python .\demo.py
please input k:(suggest 9 , 10, 11)
11
```

The detailed assignment results(seq_id_k.map) and the statistical results (count_k.txt) are in demo_result folder

K = 9

(1) Total number of short sequences: 1876

(2) The number of reads that can be assigned: 1663

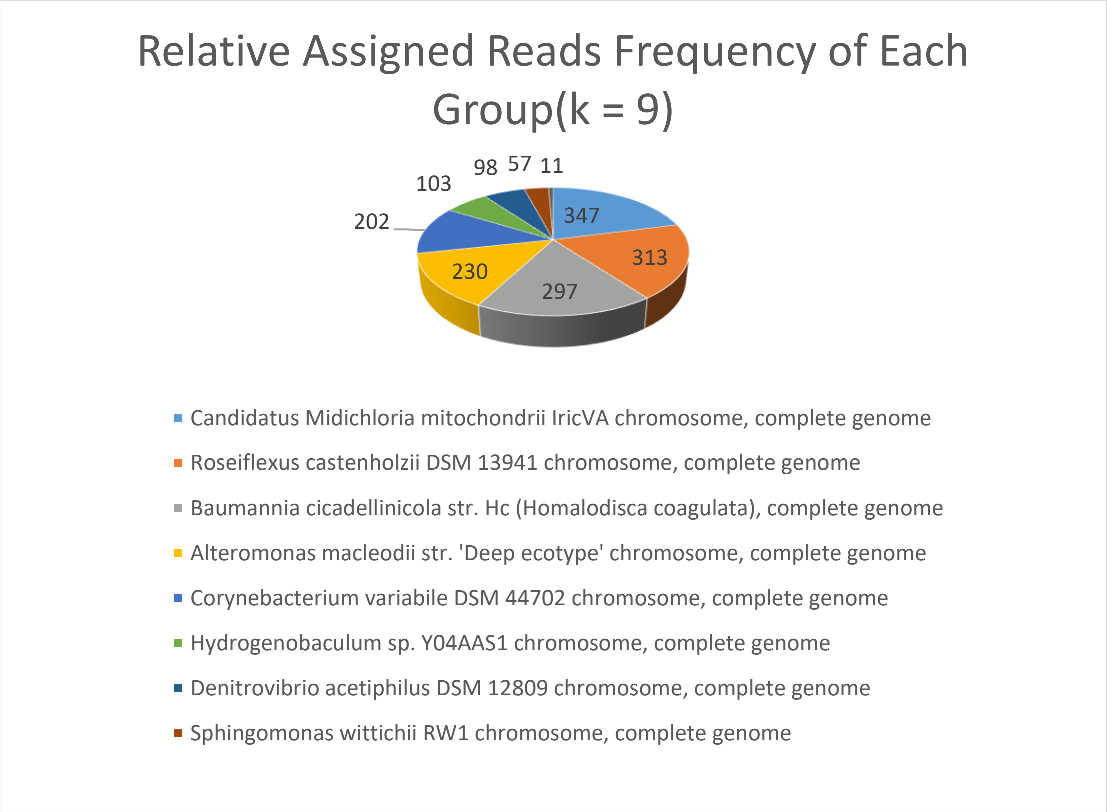
(3) The number of reads that can’t be assigned: 213

(4) The number of groups that can be assigned 1,5,10,50 reads

Minimum number of short sequences in a group	Number of groups
1	10
5	10
10	9
50	8

(5) Reads number in each group(number > 10 listed)

Reads number in each group(number > 10 listed)	
Candidatus Midichloria mitochondrii IricVA chromosome, complete genome	347
Roseiflexus castenholzii DSM 13941 chromosome, complete genome	313
Baumannia cicadellinicola str. Hc (Homalodisca coagulata), complete genome	297
Alteromonas macleodii str. 'Deep ecotype' chromosome, complete genome	230
Corynebacterium variabile DSM 44702 chromosome, complete genome	202
Hydrogenobaculum sp. Y04AAS1 chromosome, complete genome	103
Denitrovibrio acetiphilus DSM 12809 chromosome, complete genome	98
Sphingomonas wittichii RW1 chromosome, complete genome	57
Psychromonas ingrahamii 37 chromosome, complete genome	11



Relative Assigned Reads Frequency of Each Group(k = 9)

K = 10

(1) Total number of short sequences: 1876

(2) The number of reads that can be assigned: 1850

(3) The number of reads that can’t be assigned: 26

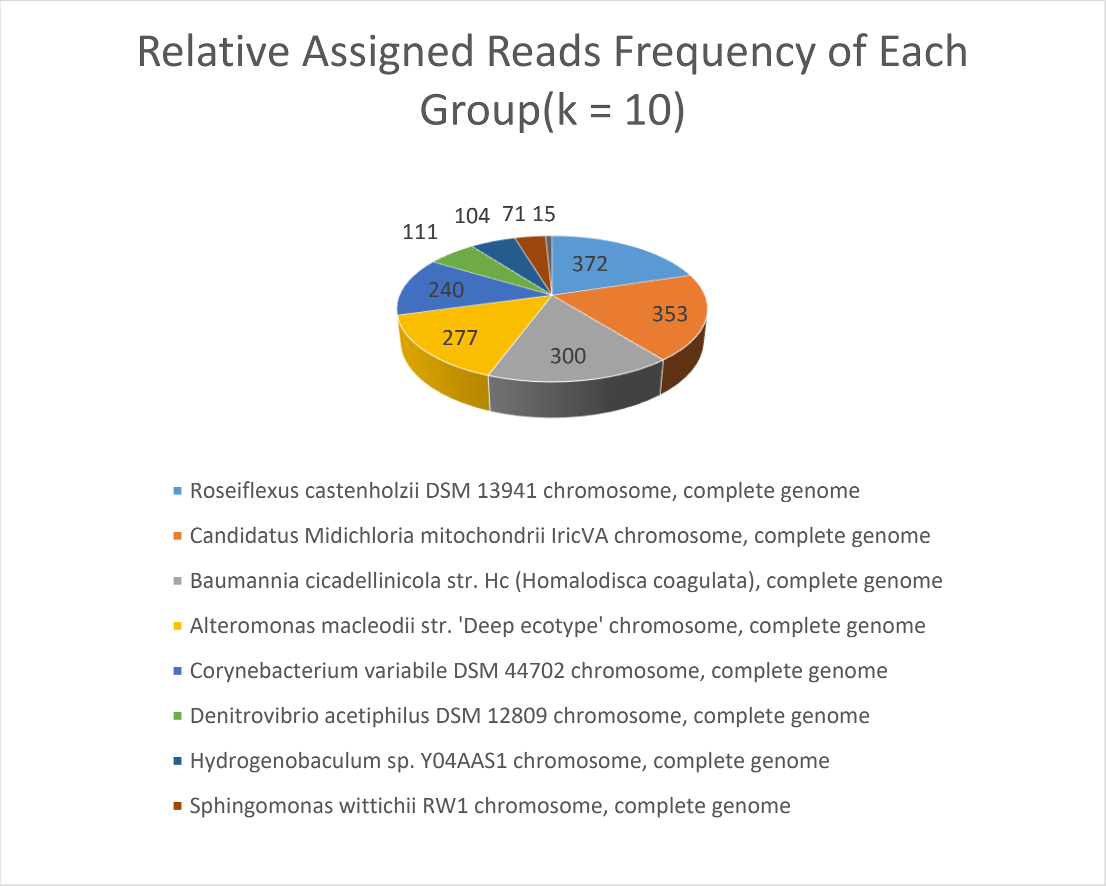
(4) The number of groups that can be assigned 1,5,10,50 reads

Minimum number of short sequences in a group	Number of groups
1	10
5	10
10	9
50	8

(5) Reads number in each group(number > 10 listed)

Reads number in each group(number > 10 listed)	
Roseiflexus castenholzii DSM 13941 chromosome, complete genome	372
Candidatus Midichloria mitochondrii IricVA chromosome, complete genome	353
Baumannia cicadellinicola str. Hc	300

(Homalodisca coagulata), complete genome	
Alteromonas macleodii str. 'Deep ecotype' chromosome, complete genome	277
Corynebacterium variabile DSM 44702 chromosome, complete genome	240
Denitrovibrio acetiphilus DSM 12809 chromosome, complete genome	111
Hydrogenobaculum sp. Y04AAS1 chromosome, complete genome	104
Sphingomonas wittichii RW1 chromosome, complete genome	71
Psychromonas ingrahamii 37 chromosome, complete genome	15



K = 11

(1) Total number of short sequences: 1876

(2) The number of reads that can be assigned: 1876

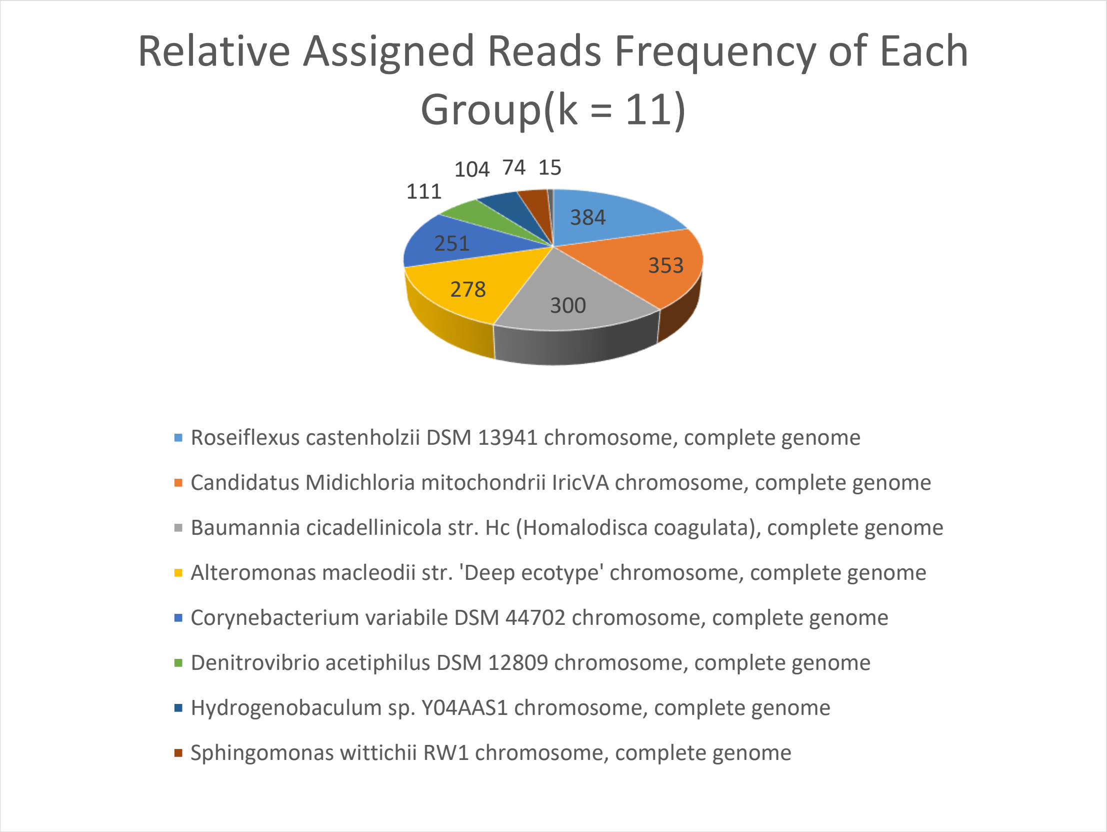
(3) The number of reads that can’t be assigned: 0

(4) The number of groups that can be assigned 1,5,10,50 reads

Minimum number of short sequences in a group	Number of groups
1	10
5	10
10	9
50	8

(5) Reads number in each group(number > 10 listed)

Reads number in each group(number > 10 listed)	
Roseiflexus castenholzii DSM 13941 chromosome, complete genome	384
Candidatus Midichloria mitochondrii IricVA chromosome, complete genome	353
Baumannia cicadellinicola str. Hc (Homalodisca coagulata), complete genome	300
Alteromonas macleodii str. 'Deep ecotype' chromosome, complete genome	278
Corynebacterium variabile DSM 44702 chromosome, complete genome	251
Denitrovibrio acetiphilus DSM 12809 chromosome, complete genome	111
Hydrogenobaculum sp. Y04AAS1 chromosome, complete genome	104
Sphingomonas wittichii RW1 chromosome, complete genome	74
Psychromonas ingrahamii 37 chromosome, complete genome	15



4. Discussion:

As discussed in the test step, the bigger the k, the more accurate the assignment is. I think this is because bigger k means bigger distinction in different matrixes. We can see that from the enlarged score gap between the top score and the second score if I print them out.

A high order markov model is usually taken as a one order markov model. This is because the fundamental probabilistic theory. For instance, a sequence is ATGCATGC, k =3, then P(C|ATG) = P(TGC|ATG). So if I want to use maximum likelihood to calculate the transition probability between ATG and TGC, I can use C and ATG to supersede it.

5. Codes:

Codes are split into 3 files, test.py, compare.py and demo.py. All these files are listed in the attachment.